基于随机投影与集成学习的离群点检测算法 *

郭一阳 [1a]，于 炯 [1a, 1b†]，杜旭升 [1a]，曹 铭 [2]

(1. 新疆大学 a. 信息科学与工程学院; b. 软件学院, 乌鲁木齐 830091; 2. 中国海洋大学 信息科学与工程学院, 山东 青岛 266100)

摘 要：针对传统基于相似度的离群点检测算法在高维不均衡数据集上效果不够理想的问题，文中提出一种新颖的基于随机投影与集成学习的离群点检测(ensemble learning and random projection-based outlier detection，EROD)框架。算法首先集成多个随机投影方法对高维数据进行降维，提升数据多样性；然后集成多个不同的传统离群点检测器构建异质集成模型，增加算法鲁棒性；最后使用异质模型对降维后的数据进行训练，训练后的模型经过两次优化组合以降低泛化误差，输出最终的对象离群值，离群值高的对象被算法判定为离群点。分别在 4 个不同领域的高维不均衡真实数据集上进行对比实验，结果表明该算法与传统离群点检测算法和基于集成学习的离群点检测算法相比，在 AUC 和 Precision@n 值上平均提高了 3.6%和 14.45%，证明 EROD 算法具有处理高维不均衡数据异常的优势。

关键词：数据挖掘；离群点检测；随机投影；集成学习

中图分类号：TP311.1 doi: 10.19734/j.issn.1001-3695.2022.02.0053

Outlier detection algorithm based on random projection and ensemble learning

Guo Yiyang[1a], Yu Jiong[1a, 1b†], Du Xusheng[1a], Cao Ming[2]

(1. *a. College of Information Science & Engineering*, *b. School of Software*, *Xinjiang University*, *Urumqi* 830091, *China*; 2. *Ocean University Of China*, *College of Information Science & Engineering*, *Qingdao Shandong* 266100, *China*)

Abstract: To address the problem that traditional similarity-based outlier detection algorithms were not effective enough on high-dimensional unbalanced datasets, this paper proposed a novel Ensemble learning and Random projection-based Outlier Detection (EROD) framework. Firstly, the EROD algorithm integrated several random projection methods to reduce the dimensionality of high-dimensional data, which improved the data diversity. Secondly, it integrated several different traditional outlier detectors to build a heterogeneous ensemble model, which increased the robustness of the algorithm. Finally, the EROD acquired the final outlier value of the object by using the heterogeneous ensemble model to train the reduced-dimensional data and by using two optimal combinations of the trained model to reduce the total error, and the algorithm determined the object with high outlier value as outlier point. The results showed that the algorithm had an average improvement of 3.6% and 14.45% in AUC and Precision@n value compared with the traditional outlier detection algorithm and the outlier detection algorithm based on ensemble learning. Therefore, the EROD algorithm has the advantage of handling the anomalies of high-dimensional unbalanced data.

Key words: data mining; outlier detection; random projection; ensemble learning

0 引言

与正常数据相比，离群点是具有不同特征的数据点，其被定义为：假设某一个数据在数据集中远远地偏离其他绝大多数数据，那么该数据被认知为与其他数据所产生的机制不相同，则它被判定为离群点[1]。之所以删除离群点是数据挖掘中不可或缺的预处理步骤，是因为离群点的存在对数据统计分析的结果有严重的负面影响[2]。因此，为了删除离群点，首先需要对其进行识别，这是离群点检测算法的首要目标。

离群点检测是一项重要的机器学习任务，它可以在具有许多高风险应用的常规数据对象中检测出异常对象，例如：流量反作弊检测。

据《2020 年中国异常流量报告》，异常流量约占整体的 8.6 个百分点。作为全球最大的广告流量平台，阿里妈妈(隶属于阿里巴巴集团)拥有超过 1000 亿美元的商业流量，这代表着其为黑灰产业瞄准的首要对象。从阿里妈妈团队的业务角度分析，流量反作弊检测的核心思想之一是识别欺诈和低质量的异常流量内容，以保护客户和平台的权益。在当前的机器学习领域，流量反作弊检测可能是对算法鲁棒性和解释性要求最高、精确度要求最高、系统规模和时效性要求最高、行业规模最大的业务。因此，流量反作弊检测技术团队必须要有"铁打"的营盘，才能够将离群点检测技术与流量反作弊应用结合得更加紧密。在流量反作弊检测任务中，高维不均衡数据的离群点检测成为国内外相关团队关注的首要焦点。

基于相似度的离群点检测算法是常见的传统无监督机器学习算法，但该种类离群点检测算法在检测高维数据时由于在距离计算方面面临维度灾难的挑战，使得难以衡量对象在高维空间分布模式上的相似度，进而导致其在检测高维不均衡数据集时，存在检测率低、参数敏感性高等问题。在现实工业界实际环境中，在没有真实的数据标签的情况下，工程师们通常要构建大量的、无监督的异质集成模型，即具有不同超参数的不同算法的集成模型，以便进一步地组合进行研

收稿日期：2022-02-10；**修回日期：**2022-04-07 **基金项目：**国家自然科学基金资助项目(61862060，61462079，61562086，61562078)

作者简介：郭一阳(1996-)，男，山东滕州人，硕士研究生，主要研究方向为机器学习、数据挖掘；于炯(1964-)，男(通信作者)，北京人，教授，博导，博士，主要研究方向为分布式计算、机器学习、数据挖掘(yujiong@xju.edu.cn)；杜旭升(1995-)，男，甘肃庆阳人，博士研究生，主要研究方向为机器学习、数据挖掘；曹铭(1996-)，女，山东菏泽人，硕士研究生，主要研究方向为机器学习、数据挖掘.

究分析，而不是依靠单个算法。因此，本文提出了一种基于随机投影和集成学习(ensemble learning and random projection-based outlier detection, EROD)的离群点检测算法。

为了提升传统的离群点检测算法在高维不均衡数据集上的检测正确率，EROD 算法应用随机投影对待检测的数据集进行降维，集成传统的离群点检测算法对降维后的数据计算出所有数据对象的离群值，通过对传统的离群点检测算法进行动态分组与优化组合，组合后的离群值作为算法最终判定的离群值。在 UCI(University of California，Irvine)真实数据集上的实验表面，EROD 算法与其他离群点检测算法对比，检测率得到了明显的提升。

本文的主要贡献总结如下：

a) 提出了一种新的无监督离群点检测框架，在数据和模型上进行了异质集成。对随机投影法进行集成以提升数据多样性，集成传统的离群点检测算法以提升模型多样性，通过两个阶段的组合，提升整体框架的检测率。

b) 针对传统的离群点检测算法在不同的高维不均衡数据集上存在不稳定性，利用集成的特性对传统算法进行均衡处理，使得整体趋于稳定，提升检测率。

c) 对传统的离群点检测算法进行了全面的参数敏感性分析，预测了整体框架的参数与性能，并对特别的数据集进行了可视化分析论述。

1 相关工作

从 19 世纪，研究学者们就已经展开了对离群点检测的科学研究[3]。基于统计与概率的离群点检测方法是一种较早提出的研究方法，这种方法根据统计与概率学原理进行检测离群现象，具有时间复杂度低的优点。其核心思想是：首先，估计出数据集的分布模型；然后，假设其中的数据对象满足该分布模型的分布规律；最后，通过评判数据对象与该分布模型是否一致来检测出数据集中存在的离群点。文献[4]受到统计函数 Copula 函数的启发，通过利用 Copula 函数预测每个给定样本的尾部分布概率，以确定其离群程度但是该种方法需要预先准确地计算出分布模型的参数，但如果不能预先准确地估计出该参数，那么将导致该种方法得到的参数估计值与真实值之间存在显著差异，使得离群点检测的准确率大幅度降低。

由于基于统计与概率的离群点检测方法的局限性，基于相似度的离群点检测研究方法在二十一世纪初被提出。基于相似度的离群点检测方法针对正常点和离群点在数据集中分布不同的特点，通过度量数据对象之间的相似度(如：距离、密度、角度等)进行检测离群点。在文献[5]中，k 最近邻(k Nearest Neighbors, kNN)、k 最近邻平均数(Average k Nearest Neighbors, Avg-kNN)和 k 最近邻中位数(Median k Nearest Neighbors, k-Median)通过计算样本之间的欧式距离来检测离群点，但它们对参数设置非常敏感，且检测高维数据时检测率低。文献[6]中提出了首个基于密度的聚类局部离群因子(Local Outlier Factor, LOF)检测方法，该技术为每个数据对象分配一个离群因子，解决了把离群值看做二元属性的问题，但无法处理多粒度和超参敏感性问题。Tang 等人[7]对 LOF 进行改进，提出了基于连接的离群因子(Connective-based Outlier Factor, COF)，该方法通过计算连接距离作为最短路径以估计邻居的局部密度，其关键思想是基于低密度和孤立性之间的区分，但是该方法与 LOF 相比耗费更多的计算成本。文献[8]一文中提出了基于角度的离群点检测(Angle-Based Outlier Detection, ABOD)方法，通过将加权余弦分值与所有近邻点的方差作为离群分值，该方法的决策边界比较复杂，容易导致过拟合。

从阅读相关文献获知，集成学习的不同的基检测器各自产生独立误差，对多个基检测器进行组合，可以在一定程度上缓解单一基检测器的超参数敏感、训练难度大和拟合效果差等问题[9]。文献[10]一文中提出了特征装袋(feature bagging, FB)离群点检测算法，该方法通过分离原始特征并创建随机的特征子集，并合并多个算法应用于该子集产生相应的离群分数，该算法提高了检测性能，但由于其检测器为同质检测器，这导致了其方法不够多样性；文献[11]文中提出了一种轻量级异常在线检测(lightweight on-line detector of anomalies, LODA)方法，其通过识别偏离大多数特征的数据进而检测出离群点，该算法有着较低的时间复杂度，但由于单个检测器输出结果不稳定导致其检测率比较低；文献[12]提出孤立森林(isolation forest, IForest)算法，其集成多棵孤立树并记录这些孤立树的路径长度，以此作为计算离群分值的依据，但若是离群点样本占比较高，与该算法所假设的离群点易被孤立的理论基础互相冲突，致使产生不理想的结果。

可以看出，基于集成学习的离群点检测算法可以通过侧重于结合模型的输出结果以生成稳定的集成模型，进而有效检测离群点。这为本文解决上述基于相似度的离群点检测算法局限性提供了思路，即 EROD 算法。EROD 算法利用集成学习的特性对传统算法进行均衡处理，且在组件检测器的选择上的理论基础互相补充，并提高了算法的鲁棒性。同时，EROD 算法在数据和模型上进行了异质集成，提升了整体结构的多样性，并通过两个阶段的组合，提升了算法的检测率。

因此，与上述离群点检测算法相比，EROD 具有鲁棒性更强、检测率更高以及不依赖先前假设的优势。

2 本文方法与理论性质

本节首先给出基于随机投影与集成学习的EROD离群点检测算法的框架与流程，然后介绍集成随机投影法，异质集成模型以及二阶段聚合算法，表 1 详细列出了本文后面内容所需的部分符号定义。

表 1 符号定义

Tab. 1 Definition of symbols

符号	定义
m	组件检测器数量
A	随机投影矩阵
X	原始数据集
Y_i	使用 m 个 A 对 X 进行随机投影后生成的 m 个数据集，i=1, 2, …, m
y_j	Y_i 中第 j 个数据
Y	Y_i 的集合
D_i	检测 Y_i 的第 i 个组件检测器
D	D_i 的集合
$D_i(y_j)$	y_j 在第 i 个组件检测器上的离群值
OF	Y_i 的离群值矩阵,其组成元素为 $D_i(y_j)$
ZOF	对 OF 进行归一化处理后的离群值矩阵
row	ZOF 行数
$outlierScore$	EROD 算法最终判定的对象离群值集合

2.1 EROD 算法整体框架与流程

EROD 算法分为 3 个步骤实现：

a) 降维。主要利用随机投影法将高维数据随机投影成低维数据；

b) 构建组件检测器集成模型。为了增强 EROD 算法的鲁棒性，将不同类别的离群点检测模型进行异质集成；

c) 二阶段聚合。将异质集成中多个组件检测器随机划分成多个不同的集群，在不同的集群中选取每个集群中的最大值，对多个最大值求均值，该均值作为EROD判定的离群

值。EROD 算法整体框架与流程如图 1 所示。

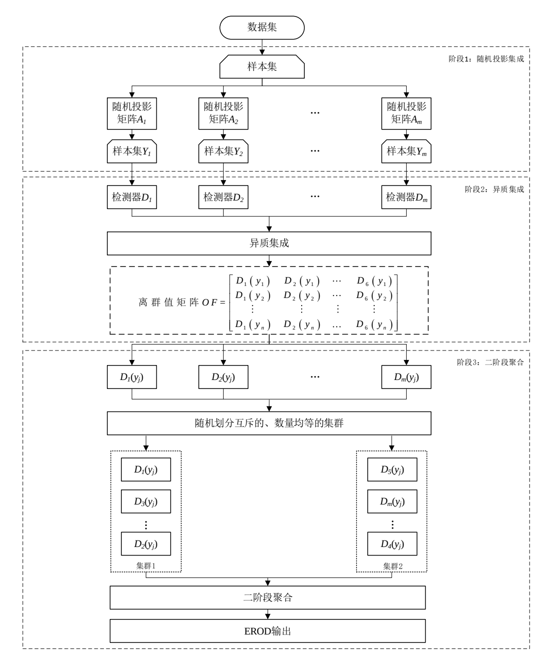


图 1　EROD 算法框架与流程

Fig. 1　Framework and process for EROD algorithm

2.2　随机投影集成

在离群点检测过程中，绝大多数离群点检测算法在高维数据上易受到维度灾难的严重影响[13]。为了解决该问题，JL 随机投影法被广泛使用进行消除维度灾难所带来的负面效果。JL 随机投影是一种降维算法，它之所以被广泛使用在离群点检测上面，是因为其降维机制可保持两两数据之间的相对距离，对高维数据在欧氏空间上进行低失真的压缩，离群点的信息在压缩过程中得以保留下来。更为重要的是，JL 随机投影法的随机机制可增强集成学习的多样性。

JL 随机投影的目的是近似保距，其理论基础是 Johnson-Lindenstrauss 辅助定理[14]。如式(1)所示，JL 随机投影表示一种线性映射关系 f：$R^d \to R^k$，即将 d 维数据随机投影为 k 维数据；如式(2)所示，由 Johnson-Lindenstrauss 辅助定理可知，$1 \le i \ne j \le n$，$\varepsilon \in (0,3)$，要以较高的概率 P 满足两两数据对象之间的相对距离保持在$(1-\varepsilon,\ 1+\varepsilon)$内，需将数据对象降维到 $k=O(\log(n)/\varepsilon^2)$维。

$$f(x_i)=x_iA, A\in R^{d\times k} \tag{1}$$

$$P\left[(1-\varepsilon)\|x_i-x_j\|^2 \le \|f(x_i)-f(x_j)\|^2 \le (1+\varepsilon)\|x_i-x_j\|^2\right] \le 2e^{-\varepsilon^2\frac{k}{6}} \tag{2}$$

如表 2 所示，根据文献[15]中的 4 种广泛使用的随机矩阵 A，可将 JL 随机投影划分为 4 种方法。

在 4 种 JL 随机投影法中，稀疏随机投影法在时间效率上略优于另外 3 种随机投影法[16]，故 EROD 算法采用稀疏随机投影法。

如式(3)(4)所示，原始数据集 X 的特征空间是由 n 个具有 d 维特征的数据构成；稀疏随机矩阵 A 是 m 个不同的稀疏随机投影矩阵，每个稀疏随机投影矩阵$\in R^{d\times k}$，Y_i 是由稀疏随机矩阵 A 作用在原始数据集 X 上得到的具有 n 个 k 维特征的数据，其中：$0<k<d$，$i=1,2,\ \dots,\ m$。

$$X=\{x_1,x_2,x_3,\cdots,x_n\}\in R^{n\times d} \tag{3}$$

$$Y_i=<X,\ A>\in R^{n\times k} \tag{4}$$

EROD 算法使用 JL 随机投影法进行集成，其基本过程如下：首先，EROD 算法使用稀疏随机投影法生成 m 个不同的稀疏随机投影矩阵 $A\in R^{d\times k}$；然后，利用这 m 个稀疏随机矩阵 A 对高维数据集 $X\in R^{n\times d}$ 进行投影，得到 m 个投影后的数据集 $Y_i\in R^{n\times k}$，最后，把 Y_i 存入集合 Y 中，输出集合 Y。

表 2　随机投影法说明

Tab. 2　Description of the random projection

JL 随机投影法	A 或 A_{ij}
高斯随机投影	A_{ij} 满足独立标准正态分布
离散随机投影	$A_{ij}=\begin{cases} \frac{1}{\sqrt{k}} & ,p=\frac{1}{2} \\ -\frac{1}{\sqrt{k}} & ,p=\frac{1}{2} \end{cases}$
循环随机投影	$A=\frac{1}{\sqrt{k}}\begin{pmatrix} b_0 & b_1 & b_2 & \cdots & b_{d-1} \\ b_{d-1} & b_0 & b_1 & \cdots & b_{d-2} \\ b_{d-2} & b_{d-1} & b_0 & \cdots & b_{d-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{d-k+1} & b_{d-k+2} & b_{d-k+3} & \cdots & b_{d-k} \end{pmatrix}\Lambda$ 其中：$b_0,b_1,\cdots,b_{d-1}$满足高斯分布；Λ为$d\times d$对角矩阵，其对角线元素满足独立伯努利分布
稀疏随机投影	$A_{ij}=\begin{cases} \sqrt{k} & ,p=\frac{1}{2\sqrt{k}} \\ 0 & ,p=1-\frac{1}{\sqrt{k}} \\ -\sqrt{k} & ,p=\frac{1}{2\sqrt{k}} \end{cases}$

具体过程如算法 1 所示。

算法 1　随机投影集成算法

输入：数据 $X\in R^{n\times d}$，数据集 X 降维后的维度 k。

输出：集合 Y。

```
a) Initialize m Sparse Random Projection matrix A={A1, A2, A3, … , Am}∈R^(d×k)  // 初始化 m 个 JL 稀疏随机投影矩阵 A
b) for Ai in A do  // 遍历 m 个 JL 稀疏随机投影矩阵 A
c) Yi=<X,Ai>∈R^(n×k)  // 对 X 进行随机投影，得到投影后的数 Yi
d) Add(Yi ,Y)  // 把数据 Yi 存入集合 Y
e) end for
f) Output(Y)  // 输出集合 Y
```

2.3　异质集成学习

EROD 离群点检测算法选择 kNN 检测器、Avg-kNN 检测器、k-Median 检测器、LOF 检测器、COF 检测器和 ABOD 检测器作为异质集成学习模型的组件检测器，即 m=6。

之所以选择这 6 种不同的离群点检测算法作为异质集成学习模型中的组件检测器，是因为相同的离群点检测算法产生的相同输出对集成学习的积极影响效果不明显[17]，换句话说，一般情况下，不同的离群点检测算法所构建成的异质集成学习模型会产生明显的积极效果。这是因为不同的组件检测器会促使集成学习在学习过程中产生多样性，可以学习数据的不同特征，进一步提升模型的泛化能力。另外，相似度高的离群点检测算法会产生相似的误差，这会对预测结果带来一定的消极影响[18]。

由于使用不同的、检出率低的离群点检测算法，虽然保证了一定的多样性，但是模型的预测率将会降低，所以应平衡多样性和检测率之间的关系。

因此，本文使用 kNN 检测器、Avg-kNN 检测器、k-Median 检测器、LOF 检测器、COF 检测器和 ABOD 检测器这 6 种具有不同特色且检测率在所有主流的离群点检测算法中较高的离群点检测算法作为异质集成学习模型的组件检测器。

如式(5)所示，异质集成学习模型中每个组件检测器对数据 Y 计算所获得的分值在此被称为离群因子 *Outlier_Factor*，每个组件组件检测器的输出为 $D(X)\in R^{n\times 1}$。

$$Outlier_Factor=[D_1(Y),\ D_2(Y),\cdots,\ D_6(Y)]\in R^{n\times 6} \tag{5}$$

异质集成基本过程如下：首先，初始化异质集成模型中的 6 个组件检测器；其次，利用初始化后的组件检测器检测

由算法 1 输出的数据 Y；最后，判定组件检测器的输出值作为数据 Y 的离群值。具体过程如算法 2 所示。

算法 2　异质集成学习算法

输入：集合 $Y=\{Y_1, Y_2, Y_3, \cdots, Y_m\}$，集合 $D=\{D_1, D_2, D_3, \cdots, D_m\}$。

输出：离群值矩阵 OF。

```
a) for i=1:Size(D) do
b) Initialize component detector Di
/* 对每个组件检测器进行初始化 */
c) end for
d) for Yi in Y do  // 遍历集合 Y
e) for yj in Yi do  // 遍历数据集 Yi
f) OF=Di(yj)   /* 利用第 i 个组件检测器检测 yj，得到 yj 的离群值 Di(yj)，将其作为离群值矩阵 OF 中的元素 /*
g) end for
h) end for
i) Output(OF)  // 输出离群值矩阵 OF
```

算法 2 中全部组件检测器在数据集 Y_i 上输出的离群值矩阵 OF 如式(6)所示。

$$OF=\begin{bmatrix} D_1(y_1) & D_2(y_1) & \cdots & D_6(y_1) \\ D_1(y_2) & D_2(y_2) & \cdots & D_6(y_2) \\ \vdots & \vdots & \vdots & \vdots \\ D_1(y_n) & D_2(y_n) & \cdots & D_6(y_n) \end{bmatrix} \tag{6}$$

离群值矩阵 OF 的物理意义：该矩阵由数据集 Y_i 中全部样本的离群因子所构成，即矩阵中的某个元素代表某个检测器对于某个样本所评估的离群程度[19, 20]。

2.4　二阶段聚合方法

如图 2 所示，偏差与方差之间存在反比关系，即随着集成学习模型复杂程度的增加，偏差下降，方差上升。这是因为复杂程度低的模型在拟合能力上是欠缺的，即组件检测器学习能力不够强，此时偏差主导了泛化误差；反之，则方差主导了泛化误差。

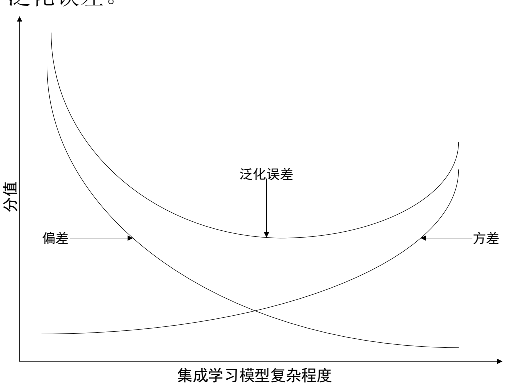


图 2　偏差-方差-泛化误差三者之间的关系

Fig. 2　The relationship among Bias-variance-total error

通常情况下，对组件检测器求均值可以达到降低方差，提高偏差的效果；对组件检测器求最大值则可以达到降低偏差，提高方差的效果。由于单一地使用任何一种组合方式可能会导致所获得的离群分值与真实分值产生较大的误差[21]。因此，合理的结合均值和最大值两种组件检测器组合方式可以起到平衡偏差与方差的作用，使得泛化误差降到一个合理的范围，提高检测率。

由于泛化误差可近似看成偏差的平方与方差之间的求和，所以，在第一阶段，对组件检测器求最大值，最大程度降低泛化误差；在第二阶段，对余下的组件检测器求均值，可使得偏差增加的幅度降到最低，进而最大程度降低泛化误差的上升幅度。

二阶段聚合基本过程：首先，对算法 2 的输出进行归一化处理，使不同离群点检测模型的输出值规范化到同一级量纲；其次，将 6 个组件检测器随机划分成 2 个集群，且每个集群中所包含的 3 个离群点检测模型存在互斥关系；最后，从每个集群中选择最大值作为该集群代表值，对每个集群代表值进行求平均，该均值作为 EROD 算法最终判定的数据对象离群值。具体过程如算法 3 所示。

算法 3　二阶段聚合算法

输入：离群值矩阵 OF。

输出：EROD 算法最终判定的对象离群值。

```
a) ZOF=Z-normalization(OF)
/* 对 OF 进行归一化处理(为避免数据表示杂乱，归一化后的数据形式仍采用表 1 中的数学符号表示) */
b) row=countRow(ZOF)  // 计算矩阵 ZOF 行数
c) for j=1:row do  // 遍历 Y1~Y6 中第 j 个数据
d) for i=1:6 do  // 遍历组件检测器
e) detectors=Di(yj)
// 将矩阵 ZOF 每行中的离群值存入集合 detectors
f) end for
g) group1,group2=randomDivide(detectors)  // 划分集群
h) max1=Max(group1)
i) max2=Max(group2)
j) outlierScore=Average(max1,max2)
k) end for
l) Output(outlierScore)
```

2.5　时间复杂度分析

设数据的数量和维度分别为 n 和 d。算法 1 中，对数据进行预处理，遍历数据进行随机投影，该阶段的时间复杂度为 O(n)；算法 2 中，使用组件检测器对数据进行计算，故该阶段的复杂度取决于组件检测器，又 COF 检测器和 ABOD 检测器都是 Fast 版本，故 kNN 检测器、Avg-kNN 检测器、k-Median 检测器、LOF 检测器、COF 检测器和 ABOD 检测器的时间复杂度分别为 O(nd)、O(nd)、O(nd)、O(n)、O(n^2)和 O(n^2)，故该阶段的时间复杂度为 O(n^2)；算法 3 中，该阶段任务是对算法 2 中的计算结果进行优化组合，该阶段的时间复杂度为 O(n)。

综上可得 EROD 算法的时间复杂度规模为 O(n^2)。

3　实验

3.1　实验环境

实验的硬件环境是：处理器为 Intel(R) Xeon(R) Gold 5117 CPU @ 2.00GHz 2.00 GHz(2 处理器)，显卡为 Nvidia Tesla V100-PCIE-16GB(共 3 块)，内存(RAM)为 256GB。

实验的软件环境是：操作系统环境为 Microsoft Windows Server 2016 Standard，算法的实现环境为 pycharm professional、python-3.6.2、tensorflow-1.14。

3.2　数据集

如表 3 所示，为了评估本文方法的检测性能，选择了 4 组均来自 UCI 数据存储库的具有不同实际应用场景的真实数据集。下面分别对该 4 组数据集的具体信息进行详细论述：

a) Arrhythmia 数据集：该原始数据集承载的是心律失常的信息，属于多类分类数据集，共 16 个类别和 279 个维度，其作用是区分是否存在心律失常现象。现对该原始数据集进行预处理，删除 5 个维度，第 3、4、5、7、8、9、14、15 等一系列小类别被定义为离群，其余类为正常。处理后的数据集总共包含 452 个样本对象，每个样本包含 274 个维度，其中有 66 个样本对象作为离群样本。

b) Mnist 数据集：该原始数据集承载的是手写数字的图像信息，包含数字 0 到 9 等 10 个图像类别。现对该原始数据集进行预处理，数字 0 被定义为正常，其余数字被定义为

离群，从原始数据集 784 个维度中随机选择 100 个维度作为处理后的样本维度。处理后的数据集总共包含 7603 个样本对象，每个样本包含 100 个维度，其中有 700 个样本对象作为离群样本。

c) Musk 数据集：该原始数据集承载的是麝香分子的信息，其作用是根据分子区分是否为麝香。现对该原始数据集进行预处理，编号 j146、j147 和 252 等非麝香类被定义为正常，编号 213 和 211 等麝香类被定义为离群，删除其他类别。处理后的数据集总共包含 3062 个样本对象，每个样本包含 166 个维度，其中有 97 个样本对象作为离群样本。

d) Speech 数据集：该数据集承载的是现实世界中语音的信息，其中美国口音占比最大，其作为正常类，其余口音被定位为离群。该数据集总共包含 3686 个样本对象，每个样本包含 400 个维度，其中有 61 个样本对象作为离群样本。

表 3　数据集信息

Tab. 3　Details of four datasets

数据集	样本数	维度	离群点数	离群点比例/%
Arrhythmia	452	274	66	15
Mnist	7603	100	700	9.2
Musk	3062	166	97	3.2
Speech	3686	400	61	1.65

3.3　评价指标

在评估检测性能和指导检测器建模时，评价指标起着不可或缺的作用。由于本文所使用的数据均为不平衡数据集，Accuracy 评价指标在数据不平衡时，其衡量结果往往是不具备参考性。在机器学习领域，对该类数据集所使用的评价指标为 AUC(Area Under Curve)和 Precision@*n*。故本文使用这两类评价指标。

AUC 是 ROC(Receiver Operating Characteristic)曲线下的面积，其分值越大，则代表算法检测性能越强。计算公式如式(7)所示。

$$\mathrm{AUC}=\sum_{i=1}^{n_+}\sum_{j=1}^{n_-}\frac{\mathrm{I}\left[d\left(x_i^+\right)>d\left(x_j^-\right)\right]+\frac{1}{2}\mathrm{I}\left[d\left(x_i^+\right)=d\left(x_j^-\right)\right]}{n_+n_-} \tag{7}$$

其中，n_+和 n_-分别表示正样本和负样本的数量，x_i 和 x_j 分别表示第 i 个和第 j 个样本，d 表示检测器，I[]表示指示函数，该函数参数为真时，值等于 1，否则等于 0。

Precision@*n* 是 Precision 指标的特殊情况，该种评价指标是在把离群点阈值设置成指定的 n 个正例时，检测器输出的 Precision 分值。计算公式如式(8)所示。

$$\mathrm{Precision}=\frac{TP}{TP+FP} \tag{8}$$

其中，*TP*(True Positive)表示离群样本被正确标记为离群样本的数量，*FP*(False Positive)表示正常样本被错误标记为异常样本的数量。

3.4　实验设计

为验证 EROD 算法将多个组件检测器集成的有效性，将本文方法与 kNN、Avg-kNN、k-Median、LOF、COF 和 ABOD 等 6 个组件检测器以及 FB、LODA 和 IForest 等 3 个集成学习算法分别进行了对比实验；同时，为保证 EROD 算法的时效性，EROD 算法与较新的同类方法 EAOD(ensemble and autoencoder-based outlier detection, EAOD)[22]和 GAN-VAE(generative adversarial network and variational auto-encoder based outlier detection, GAN-VAE)[23]在高维不均衡数据集 Mnist 上，以 AUC 值为评估指标，进行了对比实验。

在实验中，EROD 算法为了平衡维度灾难和数据多样性带来的影响，JL 随机投影将数据维度压缩为原来的三分之二。同时，为了探究 EROD 算法对其起到决定性参数的敏感程度，对集成学习中每个组件检测器的近邻参数 k 进行了敏感性实验分析，进一步地从其中选择出对 EROD 算法检测性能影响较为积极的取值参数 k，并依此建立 EROD 离群点检测模型。

在实验中，在分析并选择出对 EROD 算法检测性能影响较为积极的取值参数 k 后，对比算法 kNN、Avg-kNN、k-Median、LOF、COF 和 ABOD 的参数 k 与 EROD 中相对应的组件检测器的参数 k 保持一致；在对比集成学习算法中，FB 算法的基检测器设置为 LOF 检测器，且与 EROD 中组件检测器 LOF 的参数保持一致；LODA 算法中参数为自动优化；IForest 算法的采样大小参数 Ψ设置为 256 和树的数目参数 tn 设置为 100；同时，为保证实验的公平性和合理性，设置 EAOD 中检测器个数与 EROD 中检测器个数等同。

为了确保本实验的结果具有稳定性，现对 EROD 算法和其对比算法分别执行 10 次，对该 10 次产生的结果计算均值作为最终的结果。

3.5　参数敏感性分析与选择

为了使用 EROD 算法进行离群点检测，本文对集成模型中各个组件检测器中的近邻参数 k 做不同的取值进行对比实验，进一步地从其中选择出对 EROD 算法检测性能影响较为积极的取值参数 k，并建立 EROD 离群点检测模型。

近邻参数 k 具体选择策略为：首先，近邻参数 k 取值范围为[10, 100]，取值间隔为 10；然后，在不同 k 值上，分析组件检测器在 Arrhythmia，Mnist，Musk，Speech 这 4 个数据集上的 4 个 AUC 分值，对该 4 个 AUC 分值取均值；最后，对计算得到的 10 个 AUC 均值取最大值，该最值对应的 k 值作为组件检测器的近邻参数理想选取值的参考依据。具体过程如算法 4 所示。

算法 4　组件检测器近邻参数 k 选择策略

```
输入：k 值初始值，组件检测器 D，数据集 Arrhythmia，Mnist，
Musk，Speech。
输出：k 值参考值。
a) k=[10,20,30,40,50,60,70,80,90,100]
b) AUC=[]
c) avgAUC=[]
d) max=0
e) j=1
f) for i=k[j],i<101,j=j+1 do
g) AUC.append(D(i, Arrhythmia))
h) AUC.append(D(i, Mnist))
i) AUC.append(D(i, Musk))
j) AUC.append(D(i, Speech))
k) avgAUC.append(Average(AUC))
l) end for
m) k_reference=Max(avgAUC)
n) output(k_reference)
```

如图 3 所示，kNN 组件检测器在 Arrhythmia，Mnist，Musk，Speech 数据集上：

从 k=10 逐次递增到 k=40 的过程中，AUC 均值处于显著上升趋势；从 k=40 逐次递增到 k=80 的过程中，AUC 均值涨幅较为微小；从 k=80 逐次递增到 k=90 的过程中，AUC 均值处于不明显下降状态；AUC 均值在 k=90 和 k=100 两处相等；当 k=80 时，AUC 均值达到最大值 0.7838。但是，从 k=40 开始，AUC 均值变化不大。因此，kNN 组件检测器在 k=40 时处于最优状态。

如图 4 所示，Avg-kNN 组件检测器在 Arrhythmia，Mnist，Musk，Speech 数据集上：

从 k=10 逐次递增到 k=100 的过程中，AUC 均值变化趋

势为上升状态。其中，从 k=10 递增到 k=50 的过程中，AUC 均值上升幅度较为明显；从 k=50 以后，AUC 均值上升幅度较小；当 k=100 时，AUC 均值达到最大值 0.7840。因此，Avg-kNN 组件检测器在 k=50 时处于最优状态。

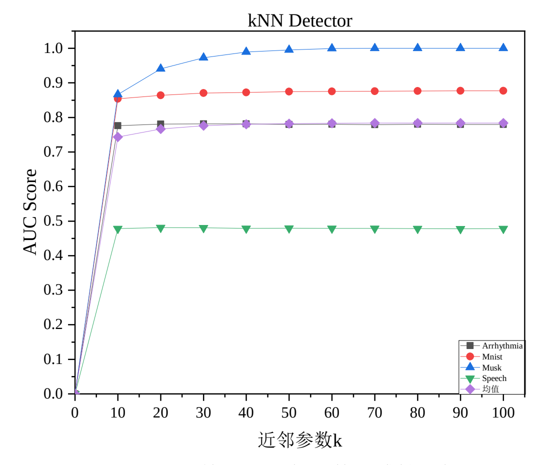


图 3 kNN 检测器近邻参数敏感性分析

Fig. 3 Sensitivity analysis of knn detector's neighbor parameters

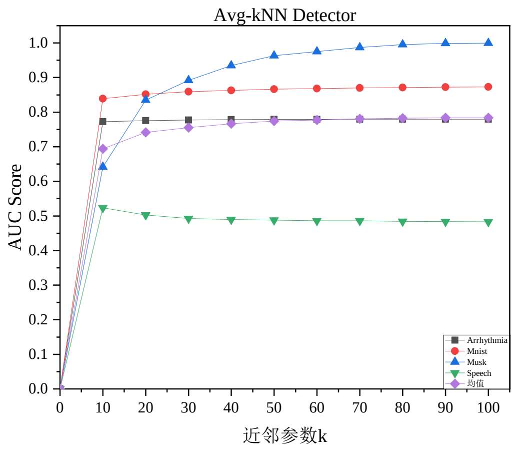


图 4 Avg-kNN 检测器近邻参数敏感性分析

Fig. 4 Sensitivity analysis of Avg-knn detector's neighbor parameters

如图 5 所示，k-Median 组件检测器在 Arrhythmia，Mnist，Musk，Speech 数据集上：

从 k=10 逐次递增到 k=100 的过程中，AUC 均值不断上升。当 k 从 10 增加至 60 时，AUC 上升较为显著；当 k 从 60 增加至 100 时，AUC 上升较为细微；当 k=100 时，AUC 均值达到最大值 0.7821。因此，k-Median 组件检测器在 k=60 时处于最优状态。

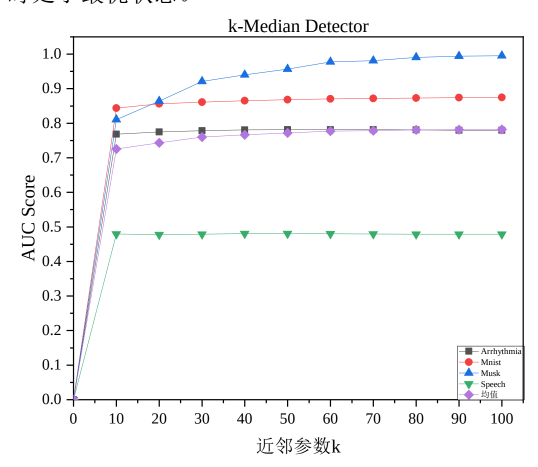


图 5 k-Median 检测器近邻参数敏感性分析

Fig. 5 Sensitivity analysis of k-Median detector's neighbor parameters

如图 6 所示，LOF 组件检测器在 Arrhythmia，Mnist，Musk，Speech 数据集上：

从 k=10 逐次递增到 k=100 的过程中，AUC 均值先上升，后下降，再上升。其中，当 k 从 10 增加到 20 时，AUC 均值显著上升；当 k 从 20 增加到 80 时，AUC 均值近似于线性下降；当 k 从 80 增加到 100 时，AUC 均值激增；当 k=100 时，AUC 均值达到最大值 0.7612。从宏观角度观察，k=100 对应的 AUC 均值明显高于其他 k 值对应的 AUC 均值。因此，LOF 组件检测器在 k=100 时处于最优状态。

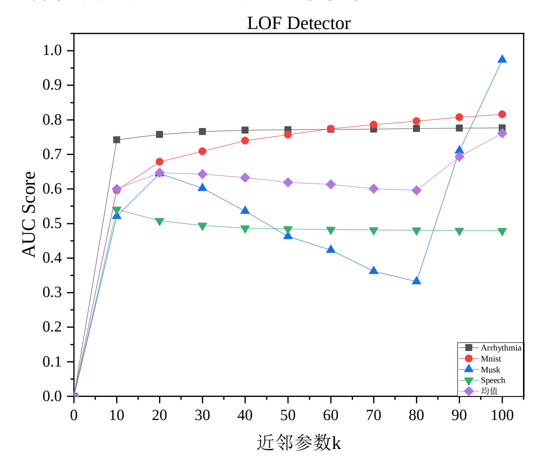


图 6 LOF 检测器近邻参数敏感性分析

Fig. 6 Sensitivity analysis of LOF detector's neighbor parameters

如图 7 所示，COF 组件检测器在 Arrhythmia，Mnist，Musk，Speech 数据集上：

从 k=10 逐次递增到 k=100 的过程中，AUC 均值先上升，后下降，其中，当 k 由 10 增加到 50 的过程中，AUC 均值处于上升状态；当 k 由 50 增加到 100 的过程中，AUC 均值处于下降状态；当 k=50 时，AUC 均值达到峰值 0.6397。因此，COF 组件检测器在 k=50 时处于最优状态。

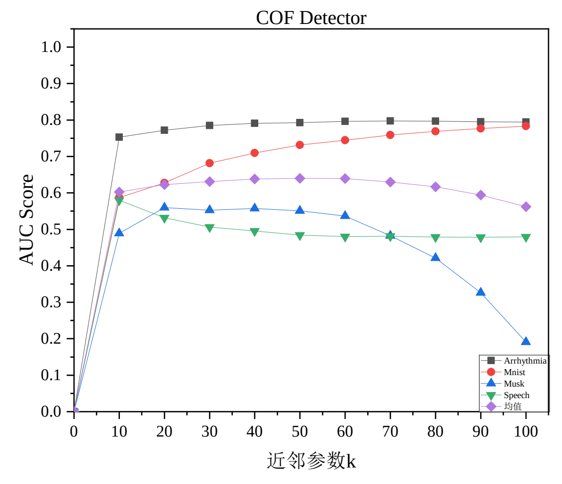


图 7 COF 检测器近邻参数敏感性分析

Fig. 7 Sensitivity analysis of COF detector's neighbor parameters

如图 8 所示，ABOD 组件检测器在 Arrhythmia，Mnist，Musk，Speech 数据集上：

从 k=10 逐次递增到 k=100 的过程中，AUC 均值先下降，在上升，但是其变化幅度十分细微。其中，k 由 10 增加到 70 时，AUC 均值以近似于水平的细微程度缓慢下降；k 由 70 增加到 100 时，AUC 均值又以近似于水平的细微程度缓慢上升；当 k=10 时，AUC 均值达到峰值 0.5807。因此，ABOD 组件检测器在 k=10 时处于最优状态。

综上所述，kNN，Avg-kNN，k-Median，LOF，COF，ABOD 这 6 个组件检测器的近邻参数 k 分别取值为 40，50，60，100，50，10 时，它们的性能处于最优。因此，选取该些近邻参数取值作为 EROD 算法中各个组件检测器的近邻参数取值。

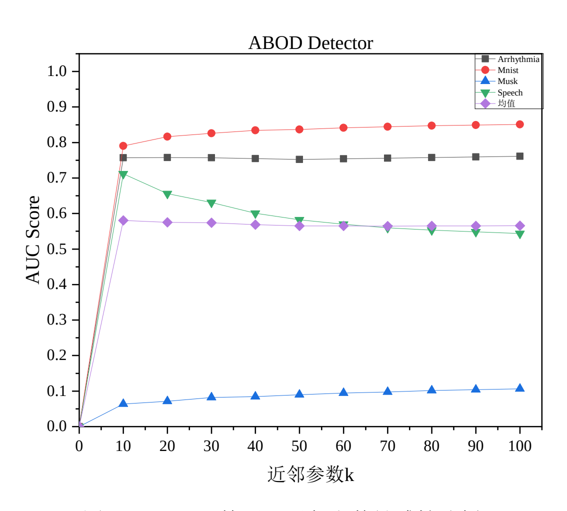


图 8　ABOD 检测器近邻参数敏感性分析

Fig. 8　Sensitivity analysis of ABOD detector's neighbor parameters

3.6　实验结果与分析

表 4 给出了在 4 个不同的高维数据集上 EROD 与 kNN、Avg-kNN、k-Median、LOF、COF 和 ABOD 的比较结果，表中加粗的数字代表检测性能最强的两个算法。而且，图 9 和 10 分别给出了在不同数据集上各算法的 AUC 分值和 Precision 分值的比较。

表 5 给出了在 4 个不同的高维数据集上 EROD 与 FB、LODA 和 IForest 等 3 个集成学习算法的比较结果，表中加粗的数字代表检测性能最强的两个算法。而且，图 11 和 12 分别给出了在不同数据集上各算法的 AUC 分值和 Precision 分值的比较。

图 13 给出了 EROD 与较新的两个同类方法 EAOD 和 GAN-VAE 在高维不均衡数据集 Mnist 上，以 AUC 值为评估指标，进行了对比实验。

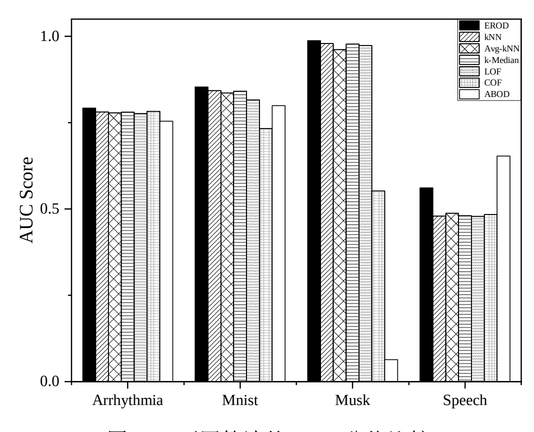


图 9　不同算法的 AUC 分值比较

Fig. 9　Comparison of AUC scores of different algorithms

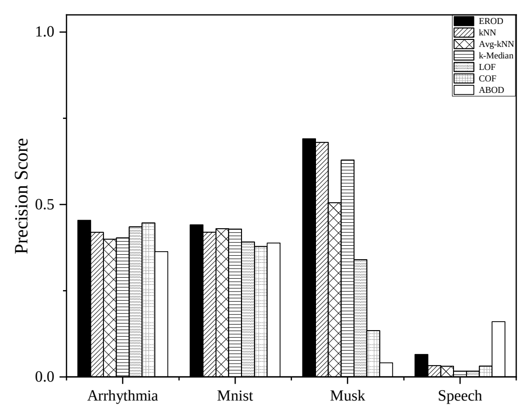


图 10　不同算法的 Precision 分值比较

Fig. 10　Comparison of Precision scores of different algorithms

表 4　EROD 算法与各组件检测器的比较

Tab. 4　Comparison of EROD algorithm with each component detector

数据集	方法	AUC Score	Precision Score
Arrhythmia	EROD	**0.7922**	**0.4545**
	kNN	0.7809	0.4191
	Avg-kNN	0.7787	0.3994
	k-Median	0.7804	0.4032
	LOF	0.7765	0.4356
	COF	**0.7829**	**0.4468**
	ABOD	0.7544	0.3636
Mnist	EROD	**0.8537**	**0.4414**
	kNN	**0.8429**	0.4200
	Avg-kNN	0.8361	**0.4300**
	k-Median	0.8407	0.4286
	LOF	0.8161	0.3914
	COF	0.7328	0.3786
	ABOD	0.7994	0.3886
Musk	EROD	**0.9878**	**0.6907**
	kNN	**0.9792**	**0.6801**
	Avg-kNN	0.9613	0.5052
	k-Median	0.9778	0.6289
	LOF	0.9733	0.3402
	COF	0.5523	0.1340
	ABOD	0.0636	0.0412
Speech	EROD	**0.5615**	**0.0656**
	kNN	0.4790	0.0328
	Avg-kNN	0.4877	0.0307
	k-Median	0.4803	0.0164
	LOF	0.4787	0.0169
	COF	0.4839	0.0315
	ABOD	**0.6530**	**0.1603**

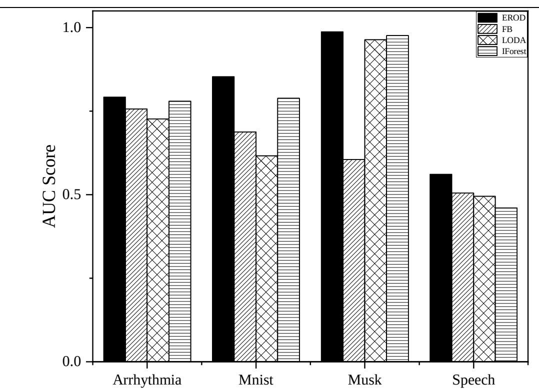


图 11　与集成学习算法的 AUC 分值比较

Fig. 11　Comparison of AUC scores of ensemble learning algorithms

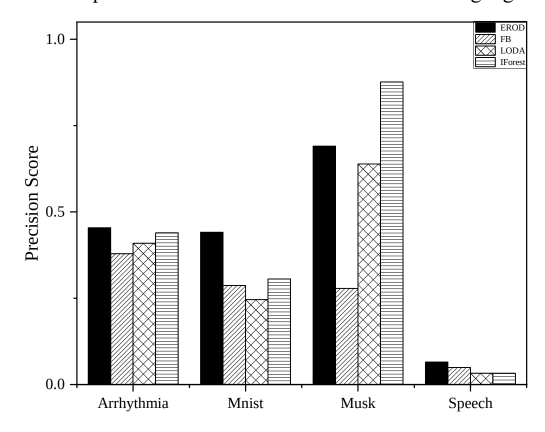


图 12　与集成学习算法的 Precision 分值比较

Fig. 12　Comparison of precision scores of ensemble learning algorithms

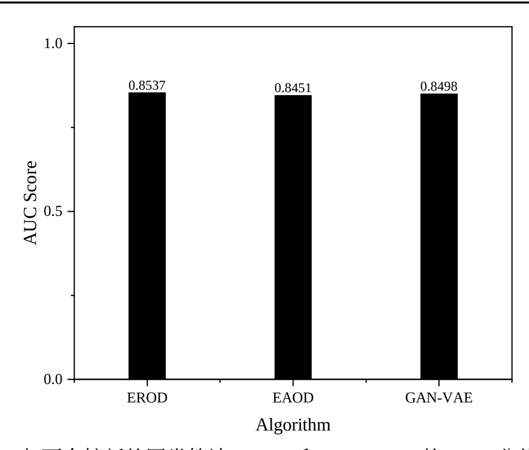


图 13　与两个较新的同类算法 EAOD 和 GAN-VAE 的 AUC 分值比较

Fig. 13　Comparison of AUC scores with two newer similar algorithms EAOD and GAN-VAE

表 5　EROD 算法与其他集成学习算法的比较

Tab. 5　Comparison of EROD algorithm with other ensemble learning algorithms

数据集	方法	AUC Score	Precision Score
Arrhythmia	EROD	**0.7922**	**0.4545**
	FB	0.7564	0.3788
	LODA	0.7271	0.4091
	IForest	**0.7798**	**0.4394**
Mnist	EROD	**0.8537**	**0.4414**
	FB	0.6875	0.2871
	LODA	0.6162	0.2457
	IForest	**0.7889**	**0.3057**
Musk	EROD	**0.9878**	**0.6907**
	FB	0.6052	0.2784
	LODA	0.9637	0.6392
	IForest	**0.9763**	**0.8763**
Speech	EROD	**0.5615**	**0.0656**
	FB	**0.5049**	**0.0492**
	LODA	0.4955	0.0328
	IForest	0.4605	0.0328

对于 EROD 算法相比较于各组件检测器，可以看出在 Arrhythmia、Mnist、Musk 上，EROD 算法的两个评价指标均优于其他算法：在 Arrhythmia 上，AUC 和 Precision 分值相较于检测性能次高的算法分别提升了 1.2 个百分点和 1.7 和百分点；在 Mnist 上，AUC 和 Precision 分值相较于检测性能次高的算法分别提升了 1.3 个百分点和 2.7 个百分点；在 Musk 上，AUC 和 Precision 分值相较于检测性能次高的算法分别提升了 0.9 个百分点和 1.6 个百分点。但是，在 Speech 上，EROD 算法的两个评价指标均处于次高状态，这是因为集成框架中大部分组件检测器在该数据集上表现较差，导致 EROD 算法平衡泛化误差的能力有所降低，但 EROD 的表现优于大部分组件检测器。

对于 EROD 算法相比较于其他集成学习算法，可以看出在 Arrhythmia、Mnist、Speech 上，EROD 算法的两个评价指标均优于其他算法：在 Arrhythmia 上，AUC 和 Precision 分值相较于检测性能次高的算法分别提升了 1.2 个百分点和 3.4 个百分点；在 Mnist 上，AUC 和 Precision 分值相较于检测性能次高的算法分别提升了 8.2 个百分点和 44 个百分点；在 Speech 上，AUC 和 Precision 分值相较于检测性能次高的算法分别提升了 11.2 个百分点和 33.3 个百分点。但是，在 Musk 上，EROD 算法在 Precision 分值上稍逊于 IForest 算法，但在 AUC 分值上均优于其他算法，相较于检测性能次高的算法提升了 1.2 个百分点，这是因为衡量指标在统计学上侧重点不同，导致 EROD 算法在 AUC 和 Precision 分值上一高一低。

对于 EROD 算法相比较于较新的同类方法 EAOD 和 GAN-VAE，在高维不均衡数据集 Mnist 上，AUC 分值分别提升了 1.02%和 0.46%，这证明了 EROD 在解决同种问题上的先进性。

在表 4 和 5 中，在 Speech 数据集上，无论是何种算法，在该数据集上分值普遍较低。如图 14 所示，Speech 在 2-D 可视化图像中，红色菱形表示离群点，其余表示正常点，可以看出这是因为在该数据集中，离群点与正常点高度地混合在一起，隐藏在正常点内部，且在维度分布上未处于尾部位置，导致其在维度分布上与正常点高度相似，使得离群点检测算法无法达到最佳检测性能。只有离群点位于暴露明显的尾部时，离群点检测算法才可精准地捕获与识别。

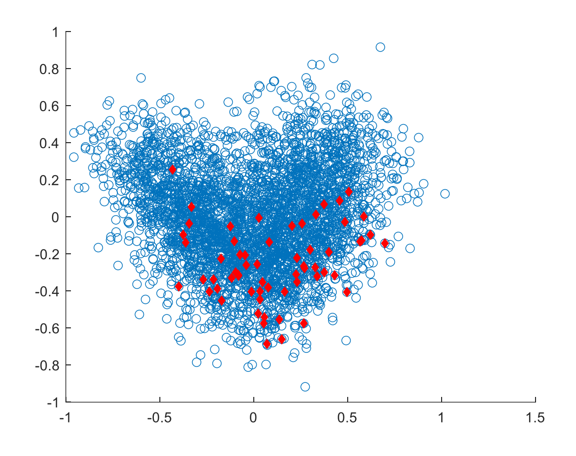


图 14　Speech 数据集 2-D 嵌入式可视化图像

Fig. 14　Speech dataset 2-D embedded visualization image

综上所述，通过与各种离群点检测算法在多个高维数据集上的对比实验，验证了 EROD 算法的有效可行性。

4　结束语

本文提出一种新的离群点检测框架-EROD，算法集成随机投影对高维数据进行降维，同时提升了数据多样性，通过对多个异质离群点检测器进行集成，提升了算法鲁棒性，之后异质集成模型对多个降维后的数据进行训练，并分两次对训练后的模型进行组合，有效降低泛化误差，提升算法检测性能。同时，从理论上分析了算法的参数敏感性，并讨论了集成组件检测器时超参的选择依据。在 UCI 数据集上实验，以 AUC 和 Precision 为评价指标对算法进行评估，与传统的离群点检测算法和基于集成学习的离群点检测算法进行比较，实验结果表明 EROD 算法具有处理高维不均衡数据异常的优势。同时，考虑到随机投影和异质检测器的集成机制对 EROD 算法效率的作用，是值得深入探讨的课题。进一步研究将从实验上研究不同的降维方式和检测器对 EROD 算法的影响以及从理论上分析 EROD 算法泛化误差临界点和其组件检测器泛化误差临界点的关系。

参考文献：

[1] Boukerche A, Zheng L, Alfandi O. Outlier detection: Methods, models, and classification [J]. ACM Computing Surveys, 2020, 53 (3): 1-37.

[2] Najafi M, He L, Philip S Y. Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization [C]// Proc of the 28th International Joint Conference on Artificial Intelligen. San Mateo, CA: Morgan Kaufmann Press, 2019: 3187-3194.

[3] Walfish S. A review of statistical outlier methods [J]. Pharmaceutical

technology, 2006, 30 (11): 82.

[4] Li Z, Zhao Y, Botta N, *et al.* COPOD: copula-based outlier detection [C]// Proc of the 20th International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2020: 1118-1123.

[5] Aggarwal C C. Outlier analysis [M]. 2nd ed. Berlin: Springer Press, 2017: 1-34.

[6] Breunig M M, Kriegel H P, Ng R T, *et al.* LOF: identifying density-based local outliers [C]// Proc of SIGMOD. New York: ACM Press, 2000: 93-104.

[7] Tang J, Chen Z, Fu A W C, *et al.* Enhancing effectiveness of outlier detections for low density patterns [C]// Proc of PAKDD. Berlin: Springer Press, 2002: 535-548.

[8] Kriegel H P, Schubert M, Zimek A. Angle-based outlier detection in high-dimensional data [C]// Proc of the 14th ACM Knowledge Discovery and Data Mining. New York: ACM Press, 2008: 444-452.

[9] Chen W, Wang Z, Zhong Y, *et al.* ADSIM: Network Anomaly Detection via Similarity-aware Heterogeneous Ensemble Learning [C]// Proc of the 17th IFIP/IEEE International Symposium on Integrated Network Management. Piscataway, NJ: IEEE Press, 2021: 608-612.

[10] Lazarevic A, Kumar V. Feature bagging for outlier detection [C]// Proc of the 11th ACM Knowledge Discovery and Data Mining. New York: ACM Press, 2005: 157-166.

[11] Pevny T. Loda: Lightweight on-line detector of anomalies [J]. Machine Learning, 2016, 102 (2): 275-304.

[12] Liu F T, Ting K M, Zhou Z H. Isolation Forest [C]// Proc of the 8th International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2008: 413-422.

[13] Pang G, Cao L, Chen L, *et al.* Learning representations of ultrahigh-dimensional data for random distance-based outlier detection [C]// Proc of the 24th ACM Knowledge Discovery and Data Mining. New York: ACM Press, 2018: 2041-2050.

[14] Cohen M B, Jayram T S, Nelson J. Simple analyses of the sparse Johnson-Lindenstrauss transform [C]// Proc of the 1st Symposium on Simplicity in Algorithms. Philadelphia, PA: SIAM Press, 2018: 1-9.

[15] Jin R, Kolda T G, Ward R. Faster Johnson–Lindenstrauss transforms via kronecker products [J]. Information and Inference: A Journal of the IMA, 2021, 10 (4): 1533-1562.

[16] Venkatasubramanian S, Wang Q. The Johnson-Lindenstrauss transform: an empirical study [C]// Proc of the 13th Workshop on Algorithm Engineering and Experiments. Philadelphia, PA: SIAM Press, 2011: 164-173.

[17] Pasillas-Diaz J R, Ratte S. An unsupervised approach for combining scores of outlier detection techniques, based on similarity measures [J]. Electronic notes in theoretical computer science, 2016, 329: 61-77.

[18] Aggarwal C C, Sathe S. Outlier ensembles: An introduction [M]. Berlin: Springer Press, 2017: 35-73.

[19] 杜旭升，于炯，陈嘉颖，等. 一种基于邻域系统密度差异度量的离群点检测算法 [J]. 计算机应用研究, 2020, 37 (07): 1969-1973. (Du Xusheng, Yu Jiong, Chen Jiaying, *et al.* Outlier detection algorithm based on neighborhood system density difference measurement [J]. Application Research of Computers, 2020, 37 (07): 1969-1973.)

[20] 杜旭升，于炯，叶乐乐，等. 基于图上随机游走的离群点检测算法 [J]. 计算机应用, 2020, 40 (05): 1322-1328. (Du Xusheng, Yu Jiong, Ye Lele, *et al.* Outlier detection algorithm based on graph random walk [J]. Journal of Computer Applications, 2020, 40 (05): 1322-1328.)

[21] Aggarwal C C, Sathe S. Theoretical foundations and algorithms for outlier ensembles [J]. ACM SIGKDD Explorations Newsletter, 2015, 17 (1): 24-47.

[22] 郭一阳，于炯，杜旭升，等. 基于自编码器与集成学习的离群点检测算法 [J/OL]. 计算机应用. [2022-03-25]. http://kns. cnki. net/kcms/detail/51. 1307. tp. 20210929. 1142. 006. html. (Guo Yiyang, Yu Jiong, Du Xusheng, *et al.* Outlier detection algorithm based on autoencoder and ensemble learning [J/OL]. Journal of Computer Applications. [2022-03-25]. http://kns. cnki. net/kcms/detail/51. 1307. tp. 20210929. 1142. 006. html.)

[23] 金利娜，于炯，杜旭升，等. 基于生成对抗网络和变分自编码器的离群点检测算法 [J]. 计算机应用研究, 2022, 39 (03): 774-779. (Jin Lina, Yu Jiong, Du Xusheng, *et al.* Generative adversarial network and variational auto-encoder based outlier detection [J]. Application Research of Computers, 2022, 39 (03): 774-779.)